PATENT ABSTRACTS OF JAPAN

(11)Publication number : **05-040463**
(43)Date of publication of application : **19.02.1993**

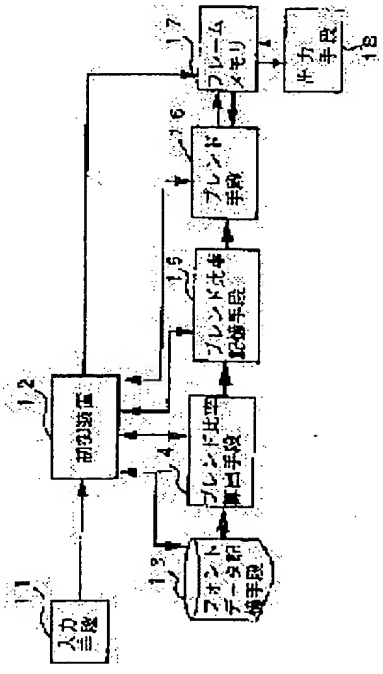
(51)Int.CI. **G09G 5/24**
G06F 3/153
G06F 15/20

(21)Application number : **03-199154**
(22)Date of filing : **08.08.1991**

(71)Applicant : **HITACHI LTD**
(72)Inventor : **YAMAZAKI NAOMI**
KAWABATA ATSUSHI

(54) MULTI-LEVEL CHARACTER GENERATOR

(57)Abstract:
PURPOSE: To output a multi-level character in which no burr is prominently observed to the whole display or a printer, etc., at high speed.
CONSTITUTION: The above purpose can be attained by providing an input means 11, a font data storage means 13, a blend ratio calculation means 14, a blend ratio storage means 15, a blend means 16, and frame memory 17, and controlling them by a controller 12. In other words, the multi-level character can be generated at high speed and outputted by finding a blend ratio without using working memory.



LEGAL STATUS

[Date of request for examination] 18.11.1994
[Date of sending the examiner's decision of rejection] 01.07.1997
[Kind of final disposal of application

other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平5－40463

(43)公開日　平成５年(1993)２月19日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
G09G 5/24		9061-5G		
G06F 3/153	310	B 9188-5B		
15/20	562	Z 7343-5L		

審査請求　未請求　請求項の数６　（全14頁）

(21)出願番号　特願平3－199154

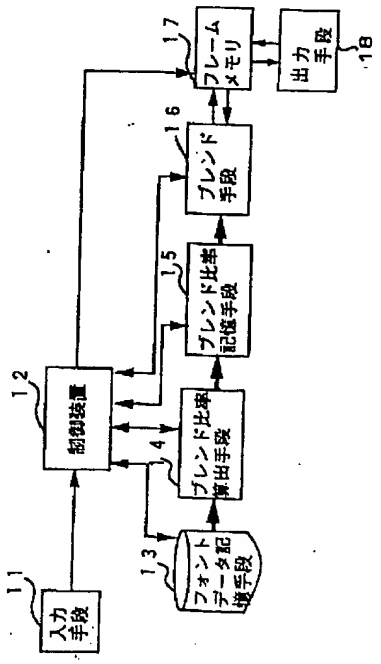
(22)出願日　平成３年(1991)８月８日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　山崎　直美
茨城県日立市久慈町4026番地　株式会社日立製作所日立研究所内
(72)発明者　川端　敦
茨城県日立市久慈町4026番地　株式会社日立製作所日立研究所内
(74)代理人　弁理士　高田　幸彦

(54)【発明の名称】多階調文字発生装置

(57)【要約】
【目的】本発明の目的は、ディスプレイ全般やプリンタなどにギザギザの目立たない多階調文字を高速に出力することにある。
【構成】入力手段(１１)フォントデータ記憶手段(１３)ブレンド比率算出手段（１４）ブレンド比率記憶手段（１５）ブレンド手段（１６）フレームメモリ（１７）を設け、制御装置（１２）で制御することによって達成される。
【効果】ワークメモリを用いずにブレンド比率を求めることにより、多階調文字を高速に生成し、出力することができる。

図１



1

【特許請求の範囲】

【請求項１】文字コードを表わす情報を入力する入力手段と、

前記文字コードを表わす情報に対応する文字の形状データを記憶するフォントデータ記憶手段と、

前記文字の形状データを構成する各ドット毎に、前記文字の形状データの領域の占める割合を数値計算によって求めるブレンド比率算出手段と、

前記ブレンド比率算出手段によって算出された前記割合に基づいて、各ドット毎に階調情報を生成するブレンド手段と、

前記ブレンド手段により生成された各ドット毎に階調情報を出力する出力手段とを具備することを特徴とする多階調文字発生装置。

【請求項２】請求項１に記載の多階調文字発生装置において、前記ブレンド比率算出手段によって算出された割合を保持するブレンド比率記憶手段を備えることを特徴とする多階調文字発生装置。

【請求項３】請求項１、または、請求項２において、前記ブレンド比率算出手段は、各ドット単位に文字部分を覆う一方向の短冊状帯領域の面積の総和を計算し、このドットにおける文字領域の占める面積と近似して、前記割合を求める手段であることを特徴とする多階調文字発生装置。

【請求項４】請求項１から請求項３の何れかにおいて、文字の存在領域を示すマスクパターンを保持する手段を備えることを特徴とする多階調文字発生装置。

【請求項５】文字コードを表わす情報を入力する入力手段と、

前記入力手段に接続され、入力された前記文字コードを表わす情報に対応する文字処理コマンドとパラメータとを生成する制御装置と、

前記文字コードを表わす情報に対応する文字の形状データを記憶するフォントデータ記憶手段と、

前記制御手段と前記フォントデータ記憶手段とに接続され、前記文字コードを表わす情報に対応する文字処理コマンドとパラメータと、前記文字コードを表わす情報に対応する文字の形状データとを受取り、前記文字の形状データを構成する各ドット毎の階調情報を生成する文字処理手段と、

前記文字処理手段に接続され、前記文字の形状データを構成する各ドット毎の階調情報を記憶するフレームバッファと、

前記フレームバッファに接続され、前記文字の形状データを構成する各ドット毎の階調情報を出力する出力手段とを具備することを特徴とするデータ処理装置。

【請求項６】請求項５において、上記文字処理手段は、単一の半導体基板に集積化された文字処理手段であることを特徴とするデータ処理装置。

【発明の詳細な説明】

2

【０００１】

【産業上の利用分野】本発明は、パソコンやワークステーションを主としたディスプレイ全般やレーザプリンタ、あるいはカラープリンタなどの多階調表示装置において、多階調文字を表示する多階調文字発生装置に関する。

【０００２】

【従来の技術】多階調文字とは、図８に示すように、文字の輪郭部分のドットを文字色と背景色との中間色で階調表示したものであり、小さな文字はきれいに、大きな文字も斜線部分のぎざぎざが目立たないように表示できる。

【０００３】従って、多階調文字であれば、描画位置を極くわずか移動しても前記説明したように文字が変形することはない。

【０００４】ところで多階調文字を生成するには、文字の輪郭部分の各ドットにおいて、ドット面積に対して文字領域の占める割合を求め、背景の色と文字の色を求めた割合に応じて混合し、文字の輪郭部分のドットの色を決定する必要がある。

【０００５】従来は、ベクトル文字情報から文字を構成する各ドットごとのブレンド比率を算出するために以下のような処理を行なっていた。

【０００６】まず、文字のベクトルデータを読み込み、展開する。次にあらかじめ膨大な計算用のワークメモリを確保しておき、その上に１個のドットを更に縦横に分割したサブピクセルパターンを考え、展開した文字を重ね合わせ、描画する。すなわち図３に示すように文字の輪郭線３３内部のサブピクセル３２を塗りつぶす。そして塗りつぶされたサブピクセル３２を数え上げてドットごとに集計し、（塗りつぶされたサブピクセルの数÷１ドット内のサブピクセルの総数）より、文字を構成するすべてのドットに対して、文字領域の占める割合（ブレンド比率）を算出する方法が知られていた。これ等に関連する技術として特開昭63－313191号公報が挙げられる。

【０００７】

【発明が解決しようとする課題】このように従来のブレンド比率算出方法では、数倍細かくしたサブピクセル単位のワークメモリ上に文字を一旦描画し、さらに描画後の状態を各ドット単位に調べ上げ、塗りつぶされたサブピクセル数を数える必要があり、処理に時間がかかるという問題があった。このとき図１５に示すように、４階調の場合には、２値の４倍，１６階調の場合には１６倍，６４階調の場合には６４倍というように、階調数が増加するにつれて、各ドットのサブピクセル数、すなわち一旦文字を描画するという処理で必要となるワークメモリが面積に比例して増加するため、処理時間も同様に増加し、膨大なワークメモリを確保しなければならず、実用的でなかった。

【０００８】上記従来技術では、文字を構成する各ドットの文字領域の占める面積割合を算出する場合に、出力する文字を数倍細かくしたサブピル単位のワークメモリ上に一旦描画するため、膨大なワークメモリを確保しなければならないという問題があった。

【０００９】また階調数によって、ワークメモリの大きさが大幅に変化するため、実用的でないという問題もあった。

【００１０】さらに描画後の状態を各ドット単位に調べ上げ、塗りつぶされたサブピクセル数を数える必要があり、描画と数え上げにかかる処理時間が長く、文字サイズ及び階調数に依存して処理時間が大幅に変化するという問題も生じていた。

【００１１】本発明は、文字を構成する各ドットの文字領域の占める面積割合を算出する場合に、ワークメモリ上に文字を一旦描画して数え上げる処理を一切行なわず、数値計算のみの処理で面積割合を算出することにより、膨大なワークメモリを確保する必要がなく、さらに描画及び１ドットずつの数え上げに要していた時間を短縮することができ、高速に多階調文字を生成することを目的としている。

【００１２】さらに、文字の存在領域を示すマスクパターンと、ブレンド比率を別に蓄え、ブレンド比率を必要最小限のドット数で管理することにより、必要なメモリ量の削減を目的としている。

【００１３】

【課題を解決するための手段】本発明によれば、各ドットごとのブレンド比率を求める際に、従来から高速化の妨げとなっていた、ワークメモリ上に文字を描画する処理や数え上げ処理を行なわずに、加算，減算，シフト演算のみの処理で済ませることにより、高速にブレンド比率を算出できる。

【００１４】また本発明によれば、ブレンド比率を一旦蓄えるブレンド比率記憶手段を備えることにより、算出したブレンド比率を再利用でき、高速に多階調文字を出力できる。

【００１５】さらに本発明によれば、ブレンド比率を一旦蓄える手段において、文字の存在領域を示すマスクパターンを、ブレンド比率とは別に蓄えることにより、ブレンド比率を管理しやすくすることができる。

【００１６】

【作用】文字を出力するためには、文字のベクトルデータの読み込み，ブレンド比率算出，ブレンドの一連の処理が必要となる。

【００１７】まずはじめに、文字のベクトルデータを読み込み、展開する。

【００１８】次にブレンド比率を算出するために、水平線ごとに、展開した文字の輪郭線との交点を検出し、数値計算のみの処理で各ドットごとに文字部分の占める面積を近似してブレンド比率を算出することにより、高速にブレンド比率を求めることができる。

【００１９】そして、前記ブレンド比率算出手段により算出されたブレンド比率を用いて、ブレンド手段に基づき、文字色として指定された色と背景色とを混合して階調色を決定し、出力する。

【００２０】その結果、従来のように出力するたびに必要とされていた各ドットごとのブレンド比率算出処理を高速に行なうことができるため、多階調文字の高速出力が可能となる。

【００２１】さらにこの応用として、ブレンド比率算出手段で求めたブレンド比率をブレンド比率記憶手段に記憶して再利用すれば、背景色及び文字色が異なっても、同じサイズの同じ文字であれば、同様なブレンド比率算出処理をしなくて済む。

【００２２】また、文字の存在領域を示すマスクパターンを、ブレンド比率とは別に蓄えることにより、ブレンド比率を管理しやすくなる。

【００２３】

【実施例】以下、本発明の１実施例を説明する。

【００２４】図１は本発明の１実施例を示すシステム構成のブロック図である。

【００２５】１１は文字コードを入力するキーボード，マウス等の入力手段であり、１２は一連の処理を制御する制御装置であり、１３はフォントデータを保存しておくフォントデータ記憶手段であり、１４は文字と多図形との色合成比率を各ドットごとに計算するブレンド比率算出手段であり、１５は前記ブレンド比率算出手段より算出した各ドットごとのブレンド比率を一旦蓄えるブレンド比率記憶手段であり、１６は前記ブレンド比率記憶手段に蓄えておいたブレンド比率に基づいて実際に合成を行なうブレンド手段であり、１７は前記ブレンド手段により生成された多階調文字を描画するフレームメモリであり、１８はフレームメモリに描画された画像情報を出力する。表示装置，プリンタ等の出力手段である。

【００２６】前記各部の動作について説明する。

【００２７】まず、出力要求された文字が入力手段１１より入力される。次に制御装置１２により、要求された文字のデータをフォントデータ記憶手段１３より読み込む。次にブレンド比率算出手段１４では、まずフォントデータを展開し、要求された文字を構成するすべてのドットにおいて文字領域の占める割合をブレンド比率として求める。次にブレンド手段１６により、文字を構成するすべてのドットに対して、各ドットごとに求めたブレンド比率に基づき、文字色及び背景色を混合して色合成を行ない、多階調文字を生成する。そしてフレームメモリ１７ではブレンド手段１６で生成した多階調文字を描画する。

【００２８】ブレンド比率記憶手段１５は、前記ブレンド比率算出手段１４で算出したブレンド比率を一旦記憶するためのものである。

【００２９】前記ブレンド比率算出手段１４では、文字を構成する各ドットにおいて、文字領域の占める割合をブレンド比率として算出し、求めたブレンド比率を０％から１００％に設定しておく。そして前記ブレンド手段１６において、このブレンド比率に基づいて文字色として指定された色と背景色とを混合して階調色を求めれば、各ドットごとに文字領域の占める割合に応じた階調色を求めることができ、特に文字の輪郭部分や斜線などは背景色を考慮した自然な色合いとなり、斜線部分のギザギザが目立たない表示結果を得ることができる。

【００３０】図９及び図１２を用いてブレンド比率算出手段１４の動作について説明する。図９はドットパターンの表示画素に対応するものであって、フォントデータ記憶手段１３には、そのドットパターン内において例えばアルファベットの「Ｎ」の輪郭座標データを含めてその他の各種の文字の輪郭座標データが記憶されている。この輪郭座標データが文字ベクトルデータである。

【００３１】まず、図９に示すように、ドットパターン上に読み込んだ文字ベクトルデータを重ね合わせた状態で、例えば図９の円内を取り出したものを図１２とすると、図１２に示すように各ドットの文字領域がその各ドット内に占める面積割合（文字領域の占める面積÷１ドットの面積）を算出する。これがブレンド比率である。

【００３２】すなわち、ドットｄ１でのブレンド比率は９８％，ドットｄ２でのブレンド比率は７５％，ドットｄ３でのブレンド比率は０％，ドットｄ４でのブレンド比率は７０％，ドットｄ５でのブレンド比率は９６％，ドットｄ６でのブレンド比率は２４％，ドットｄ７でのブレンド比率は２４％，ドットｄ８でのブレンド比率は９８％，ドットｄ９でのブレンド比率は７２％となる。

【００３３】前記ブレンド比率算出手段１４において、文字を構成する各ドットのブレンド比率を（文字領域の占める面積÷１ドットの面積）で算出すれば、各ドットごとに文字領域の占める割合を正確に反映させたブレンド比率を求めることができる。

【００３４】図２は前記ブレンド比率算出手段１４の１実施例を示す処理フローである。

【００３５】まず、ステップＳＴ２１において、文字のベクトルデータを読み込む。ステップＳＴ２２では、文字のベクトルデータを展開し、文字の輪郭線を生成する。ステップＳＴ２３では、１ドットを階調数だけ横に分割した単位で水平線を発生させ、水平線ごとに文字の輪郭線と水平線との交点を算出する。次にステップＳＴ２４において、図４に示すように、ステップＳＴ２３で算出した交点から文字領域を覆うような短冊を考える。この短冊の高さｄｙは分割数に依存しているため一律に決まる。また幅ｄｘは交点ｘ１，ｘ２を使って、ｄｘ＝ｘ２－ｘ１より容易に求まる。ステップＳＴ２５において、各短冊の面積をｄｘ×ｄｙにより算出し、ドット内における各短冊の総面積を求める。ステップＳＴ２６では（短冊の総面積÷１ドットの面積）よりブレンド比率を算出する。

【００３６】本実施例によれば、ワークメモリ上に文字を描画せずにブレンド比率を算出できるので、ブレンド比率算出時間の短縮が可能となる。さらに、膨大なワークメモリを確保する必要もない。

【００３７】また、本実施例において、１ドット全体が文字部分に覆われている場合、及び全く覆われていない場合には上記ブレンド比率を算出する処理を行なわなければ、文字の輪郭部分の処理のみで済むので高速に多階調文字を生成することができる。

【００３８】次に、図５を用いて別の実施例を説明する。

【００３９】ここでは、上述の実施例において短冊の高さｄｙが１になるように調整した場合を示す。

【００４０】ステップＳＴ５１において、文字のベクトルデータを読み込む。本実施例も前記実施例と同様に短冊を生成する。しかし本実施例の場合には、短冊の高さが１になるように、あらかじめステップＳＴ５２において、ベクトル座標データを階調数倍に拡大する。階調数は通常２，４，１６，６４，２５６等の２のべき数で表わされる。そのため、各々１，２，４，６，８ビットシフトすることにより拡大する。次にステップＳＴ５３において、拡大した文字のベクトルデータを展開し、文字の輪郭線を生成する。ステップＳＴ５４では、１ドットごとに水平線を発生させ、文字の輪郭線と水平線との交点を算出する。ステップＳＴ５５では、ステップＳＴ５４で算出した交点より文字領域を覆うような短冊を生成する。この短冊の高さｄｙはあらかじめ１になるようにステップ５２で調整してあるため、一律に１となる。また幅ｄｘは交点ｘ１，ｘ２より求まる。ステップＳＴ５６において、各短冊の面積を算出する。短冊の高さｄｙは１であるので短冊の面積は、すなわち短冊の幅となる。したがってドット内における各短冊の総面積は各短冊の幅の総和となる。ステップＳＴ５７では、はじめにベクトル座標データを階調数倍にしておいたため、（短冊の総面積÷拡大した１ドットの面積）よりブレンド比率を算出する。このとき、拡大した１ドットの面積も前記説明したように２のべき数となる。そこで同様にしてビットシフトで除算を行なう。

【００４１】この実施例によれば、あらかじめ座標データを階調数倍しておくため、短冊の面積及び総面積を求めるのに整数の加減算及びシフト演算のみの処理で済むので、計算を単純化することができ、計算時間の短縮が可能となる。

【００４２】本実施例の場合にも、膨大なワークメモリを使用する必要はない。

【００４３】これら２つの実施例は水平線ごとの処理であるため、文字サイズが大きくなっても処理時間が大幅に増加することはない。

【００４４】また、この実施例の場合にも前記実施例と同様に、１ドット全体が文字部分に覆われている場合及び全く覆われていない場合に上記ブレンド比率を算出する処理を行なわないようにすれば、高速な多階調文字生成が可能となる。

【００４５】次にこれら２つの実施例において、ブレンド比率とは別に文字の存在領域を示すマスクパターンを持つ場合について説明する。

【００４６】例えば、文字を４階調で出力する場合には、もとの１ドットの状態は図１０のうちのいずれかで表現できると考えられる。ここでは１ドットを４個のサブピクセルで表現し、各々のサブピクセルごとに文字領域とするかどうかの判定を行ない、１ドットにおける文字領域の占める割合を求めてブレンド比率とする。この場合、ドットｄ１のブレンド比率は０％，ドットｄ２のブレンド比率は２５％，ドットｄ３のブレンド比率は５０％，ドットｄ４のブレンド比率は７５％，ドットｄ５のブレンド比率は１００％となり、それぞれドットｄ１１，ドットｄ２１，ドットｄ３１，ドットｄ４１，ドットｄ５１のように示され、５通りに表現されることになる。従って、この場合ブレンド比率を管理するために必要なビット数は４階調の場合３ビットとなる。同様にして１６階調の場合５ビット，６４階調の場合７ビット，２５６階調の場合９ビットとなる。

【００４７】そこで、文字を構成する各ドットごとに文字の存在情報を管理するような１ビットのフラグを持たせる。そして例えば文字部分がドット上に少しでも存在している場合にはオン、全く存在していない場合にはオフとする。例えば図１３（ａ）の「立」の場合には図１３（ｂ）のようなマスクパターンを持たせる。同様にして図１０の場合には、ドットｄ１の文字存在情報フラグはオフ、ドットｄ２，ドットｄ３，ドットｄ４，ドットｄ５の文字存在情報フラグはオンとなる。

【００４８】このように表現すると、４階調の場合には文字存在情報フラグがオフとなるのはブレンド比率０％のときのみとなり、オンのときは２５％，５０％，７５％，１００％の４通りのブレンド比率をとることになる。ブレンド比率とは別に文字の存在領域を示すマスクパターンを持たせることによりブレンド比率だけを管理するのに必要なビット数は２ビットとなる。同様に、１６階調の場合には４ビット，６４階調の場合は６ビット，２５６階調の場合は８ビット必要になる。

【００４９】本実施例によれば、ブレンド比率を表現するためのビット数が階調数により決定できるため、必要最小限の管理情報量で済み、２５６階調の場合にもブレンド比率を高々８ビットで表現できるので、管理及びメモリ削減の面で役立つ。

【００５０】さらに、マスクパターンを持たせておけば、フラグがオフの場合には、ブレンド比率算出処理を行なう必要はない。従って、ブレンド比率算出処理が必要でないドットを見つけだすのにも便利である。

【００５１】次に、図６を用いて、別の実施例を説明する。

【００５２】本実施例は、前記実施例の応用として、一度算出したブレンド比率をブレンド比率記憶手段１５に一時記憶しておき、再利用するものである。

【００５３】まず、ステップＳＴ６１において、文字コードを指定する。ステップＳＴ６２では、要求された文字のブレンド比率がブレンド比率記憶手段１５に既に記憶されているかどうか検索する。要求された文字を構成する各ドットごとのブレンド比率が既にブレンド比率記憶手段１５に記憶されている場合には、ブレンド比率記憶手段１５に記憶されたブレンド比率情報と、背景色及び文字色より、ステップＳＴ６６で階調色を決定する。ステップＳＴ６２において、文字のブレンド比率情報がブレンド比率記憶手段１５に記憶されていない場合には、ステップＳＴ６３でベクトル文字情報から文字を構成する各ドットごとのブレンド比率を算出する。

【００５４】次にステップＳＴ６４でブレンド比率記憶手段１５に文字のブレンド比率情報を記憶する領域があるかどうかをチェックする。もし空き領域があれば、ステップＳＴ６５でブレンド比率記憶手段に文字のブレンド比率情報を記憶する。空き領域がない場合には、ステップＳＴ６８において既にブレンド比率記憶手段１５に記憶している文字の中で、最も出力要求頻度の低い文字を検索し、ブレンド比率記憶手段１５よりこの文字のブレンド比率情報を削除して新たな文字を記憶する領域を確保した後で、ステップＳＴ６５でブレンド比率記憶手段１５に文字のブレンド比率情報を記憶する。

【００５５】次に、ステップＳＴ６６で、ブレンド比率記憶手段１５に記憶されている各ドットごとの文字領域の占める割合に応じて、背景色及び文字色を混合し、各ドットの階調色を決定する。そしてステップＳＴ６７で、ブレンド手段１６で作成した多階調文字をフレームメモリ１７上に描画する。

【００５６】上記ステップＳＴ６８では出力要求頻度の低い文字を検索しているが、例えば、ブレンド比率記憶手段１５に、記憶されている文字のブレンド比率情報の使用時刻を管理しておけば、使用時刻の最も古い文字を容易に検索できる。

【００５７】つまり、ブレンド比率記憶手段１５に文字を記憶する際、及びブレンド比率記憶手段１５に記憶された文字を使用する際に、その使用時刻を記憶する。そして、文字削除を行なう際には、記憶してある使用時刻を調べ、その中で最近最も使われていない文字を削除する。

【００５８】本実施例によれば、最初に文字出力を要求されたときだけ、ブレンド比率記憶手段１５に算出したブレンド比率を記憶してしまえば、再び同じ文字を要求されたときには、それを再度利用できることにより、同

じサイズで同じ文字のブレンド比率算出処理を冗長に行なわず、高速な多階調文字出力が可能となる。

【００５９】さらに、文字のブレンド比率情報をブレンド比率記憶手段１５に記憶できなくなった場合には、出力要求頻度の低い文字をブレンド比率記憶手段１５より削除して、新たな文字を記憶することより、ブレンド比率記憶手段１５に所望の文字の存在する確率（ヒット率）を高くすることができ、より高速に多階調文字を出力できる。

【００６０】また、ブレンド比率は上記のような数値でなく、その他の表現方法であってもよいことはもちろんである。

【００６１】さらに別の実施例として、上記ブレンド比率算出処理をソフトウェアで行なわず、ＬＳＩに内蔵する場合について図７，図１１を用いて説明する。

【００６２】図７はデータ処理装置の一例となるグラフィック表示装置の構成例を示したものである。システムは好ましくは単一の半導体基板に集積化された制御装置１２，ＲＡＭ１２２，好ましくは単一の半導体基板に集積化された文字処理装置123，フォントデータ１２４，フレームバッファ１２５、及びＣＲＴ等の出力手段１７から成る。尚、図中、入力手段１１は省略されているが、図１と同様に、制御装置１２に接続されている。

【００６３】制御装置１２は、ＲＡＭ１２２に記憶されたプログラムを実行処理し、システム全体を管理制御する。すなわち、文字処理において、文字処理装置１２３に対し、文字処理コマンドとパラメータ情報を転送し、文字処理装置１２３を起動する。

【００６４】文字処理装置１２３は、加算器１２７，減算器１２８及びシフト器１２９を備えた装置であり、制御装置１２からの指示に従い、フォントデータ１２４をアクセスし、あらかじめ定められた処理手順に従ってブレンド比率を算出し、フレームバッファ１２５上での処理を行なう。

【００６５】本実施例では特に、フレームバッファ１２５には表示装置の各カラープレーンにおける各画素のブレンド比率を書き込むことにする。

【００６６】文字処理装置１２３では、前記実施例で説明した方法でブレンド比率を算出する。図５で示した一連の処理について図１８を用いて説明する。まず、フォントデータ１２４から座標データを読み込み、階調数倍に拡大する。このとき階調数は、２値，４階調，１６階調，６４階調，２５６階調と２のべき乗数(２のｇ乗)で表わせるため、全座標データをｇビットシフトして拡大する。

【００６７】次に、拡大した文字のベクトルデータを展開し、水平線ごとに文字の輪郭線と水平線との交点を算出し、ソートする。水平線ごとの交点算出は、文字の輪郭線の傾きから減算器だけを用いて容易に求めることができる。

【００６８】算出した交点ｅ１，ｅ２（ｅ１＞ｅ２）より文字領域を覆う短冊を生成する。このとき短冊の幅は２交点の差Ａｃｃで表わせる。ここではあらかじめ階調数倍に拡大してあるため、短冊の高さは１となる。従って、短冊の面積は短冊の幅Ａｃｃで示されるため、結果的に短冊の面積はＡｃｃで示され、減算器だけを用いて算出できる。

【００６９】次に１ピクセルごとに短冊の総面積を求める。これは既に求めてある各短冊の面積の総和（Brend）で求められるため、加算器だけを用いた処理となる。

【００７０】そして各短冊の面積の総和（Brend）より、背景の占める面積は（１ピクセルの面積（Gray）－Brend）となる。そこでピクセルの色は

文字色（ＲＧＢ）×Brend＋フレームメモリ（ＲＧＢ）×（Gray－Brend）

で求められる。

【００７１】ブレンド比率は（短冊の総面積÷階調数）で求められる。階調数は前記説明したように２のｇ乗と表現できるため、ｇビットシフトするだけで容易に算出できる。

【００７２】従って、ブレンド比率を求める処理は加算，減算，シフト演算のみで行なうことができる。そこで、加算器１２７，減算器１２８、及びシフト器１２９を備えた文字処理装置１２３をあらかじめ作成しておけば、ソフトウェアで処理するのに比べて、かなり高速にブレンド比率を算出することができる。

【００７３】

【発明の効果】本発明によれば、以上説明したように構成されているので、以下に記載されるような効果がある。

【００７４】各ドットごとにブレンド比率を算出する際に、数値計算のみの処理で行なうことより、高速に求めることができる。

【００７５】また、数値計算のみの処理となるため、計算用のワークメモリを使わずにブレンド比率を求めることができる。

【００７６】従って、高速にブレンド比率を算出できるため、メモリ節減及び多階調文字の高速出力が可能となる。

【００７７】また、このブレンド比率算出処理をソフトウェアで行なわず、ＬＳＩに内蔵すれば、従来と比較してかなりの高速化が可能となる。

【００７８】さらに、文字の存在領域を示すマスクパターンを、ブレンド比率とは別に蓄えることにより、ブレンド比率を管理しやすくなる。

【００７９】また、最初に文字を要求されたときだけ、算出したブレンド比率をブレンド比率記憶手段に記憶してしまえば、再び同じ文字を要求されたときに、それを再利用できるため、高速な多階調文字出力が可能とな

る。

【００８０】さらに、記憶されている文字のブレンド比率情報の使用時刻を管理しておけば、文字のブレンド比率情報を記憶する領域が無くなった場合に、出力要求頻度の低い文字を検索することができるため、常に使用頻度の高い文字を記憶しておくことができ、高速に多階調文字を出力することができる。

【図面の簡単な説明】

【図１】本発明のシステムを示すブロック図である。

【図２】本発明の１実施例の処理手順を示すフローチャートである。

【図３】従来から行なわれてきたブレンド比率算出処理を示す図である。

【図４】本発明のブレンド比率算出処理を示す図である。

【図５】本発明の別の実施例の処理手順を示すフローチャートである。

【図６】本発明の別の実施例の処理手順を示すフローチャートである。

【図７】本発明の別の実施例の構成図を示したものである。

【図８】多階調表示した場合の拡大図である。

【図９】ドットパターン上における文字ベクトルデータ図である。

【図１０】本発明の手法でブレンド比率を求める際に仮想的に示されるドットの状態を示した図である。

【図１１】本発明の別の実施例の処理手順を示すブロック図である。

【図１２】ドット内における文字領域の面積割合を示す図である。

【図１３】１実施例のマスクパターンを説明する図である。

【図１４】サブピクセルパターンのワークメモリを示したものである。

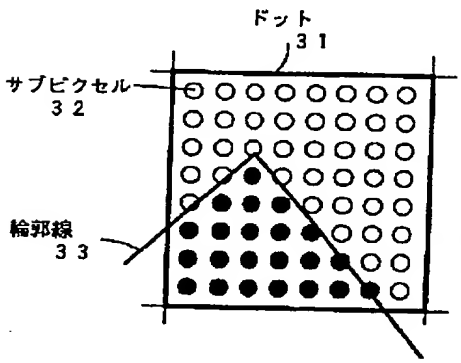
【図１５】階調数の変化に伴い、ワークメモリがどのように増加するかを示した図である。

【符号の説明】

１１…入力手段、１２…制御装置、１３…フォントデータ記憶手段、１４…ブレンド比率算出手段、１５…ブレンド比率記憶手段、１６…ブレンド手段、１７…フレームメモリ、１２２…ＲＡＭ、１２３…文字処理装置、１２４…フォントデータ、１２５…フレームバッファ、３１…ドット、３２…サブピクセル、３３…輪郭線、４１…ドット。

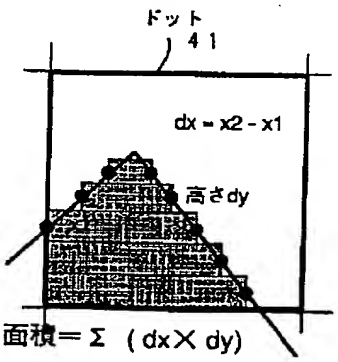
【図３】

図３



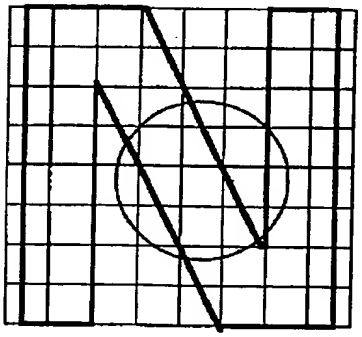
【図４】

図４



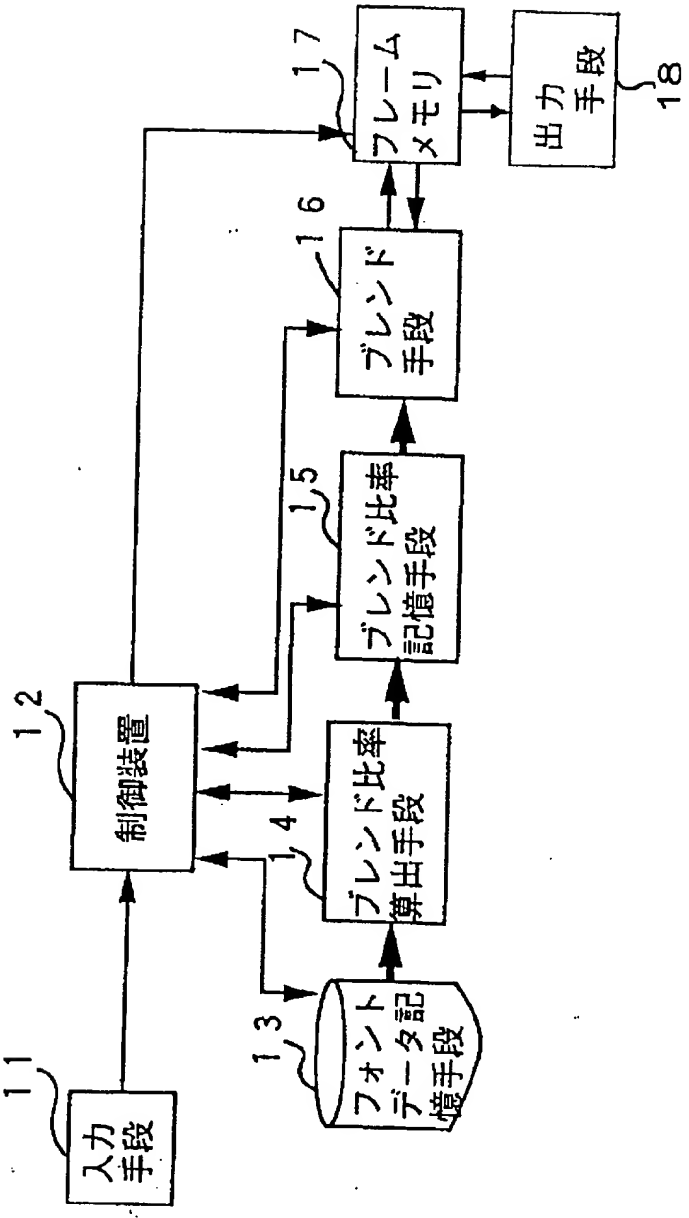
【図９】

図９



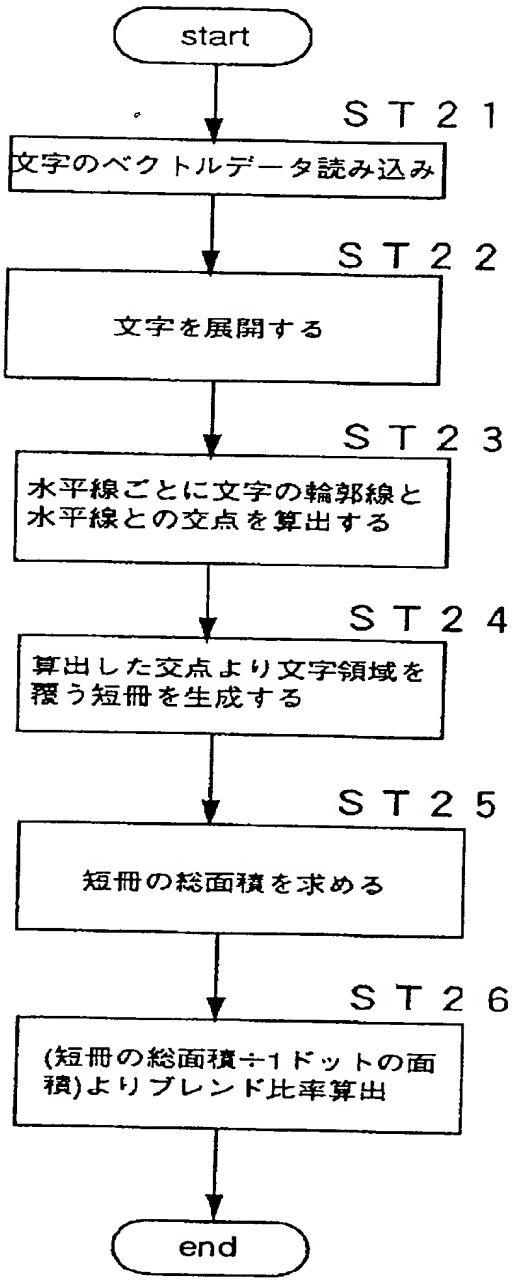
【図１】

図 1


入力手段
11
制御装置
12
フォントデータ記憶手段
13
ブレンド比率算出手段
14
ブレンド比率記憶手段
15
ブレンド手段
16
フレームメモリ
17
出力手段
18

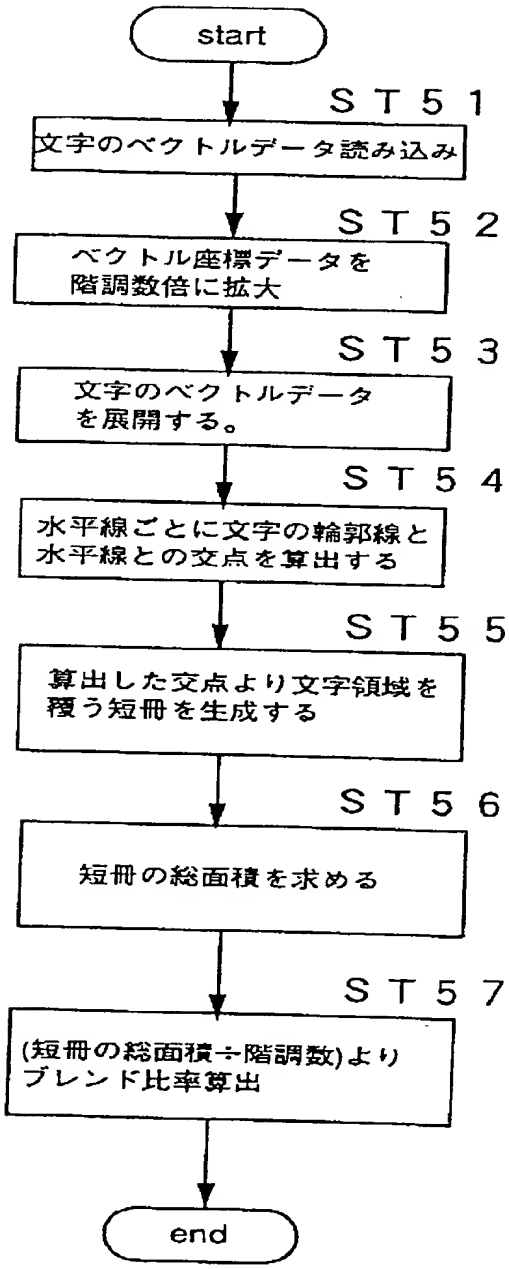
【図２】

図 ２


start
ＳＴ２１
文字のベクトルデータ読み込み
ＳＴ２２
文字を展開する
ＳＴ２３
水平線ごとに文字の輪郭線と水平線との交点を算出する
ＳＴ２４
算出した交点より文字領域を覆う短冊を生成する
ＳＴ２５
短冊の総面積を求める
ＳＴ２６
(短冊の総面積÷1ドットの面積)よりブレンド比率算出
end

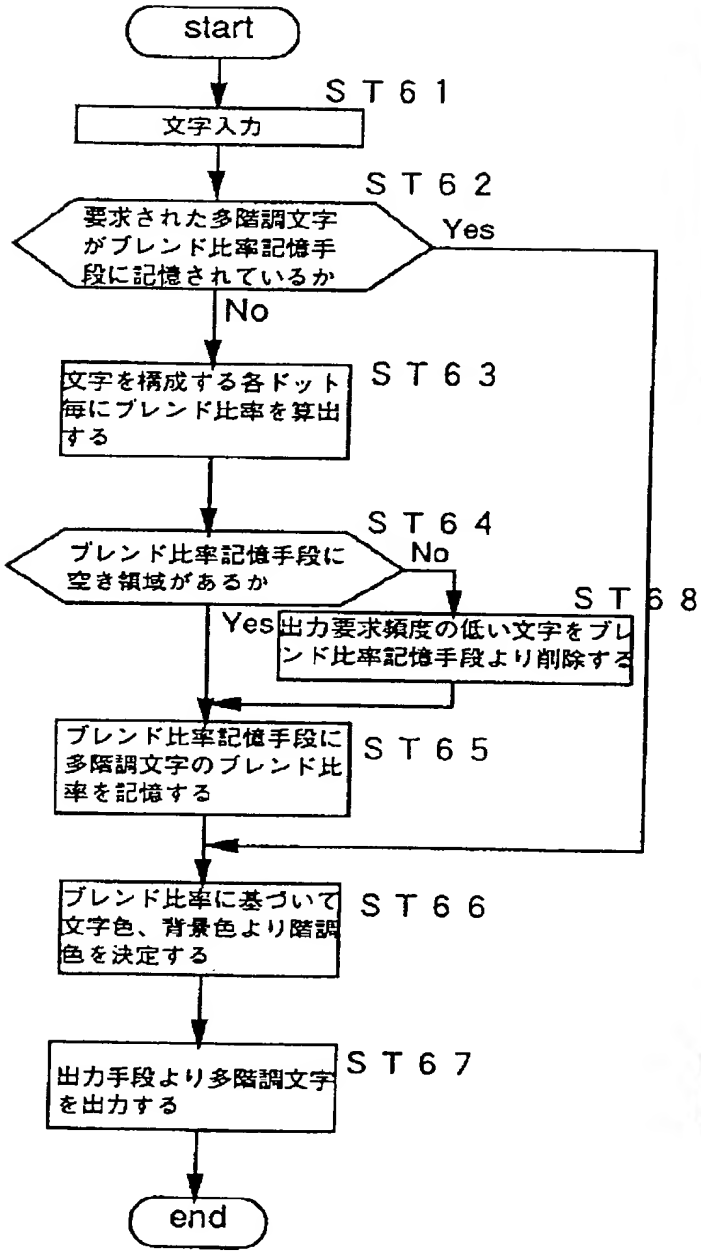
【図５】

図 ５


start
ＳＴ５１
文字のベクトルデータ読み込み
ＳＴ５２
ベクトル座標データを階調数倍に拡大
ＳＴ５３
文字のベクトルデータを展開する。
ＳＴ５４
水平線ごとに文字の輪郭線と水平線との交点を算出する
ＳＴ５５
算出した交点より文字領域を覆う短冊を生成する
ＳＴ５６
短冊の総面積を求める
ＳＴ５７
(短冊の総面積÷階調数)よりブレンド比率算出
end

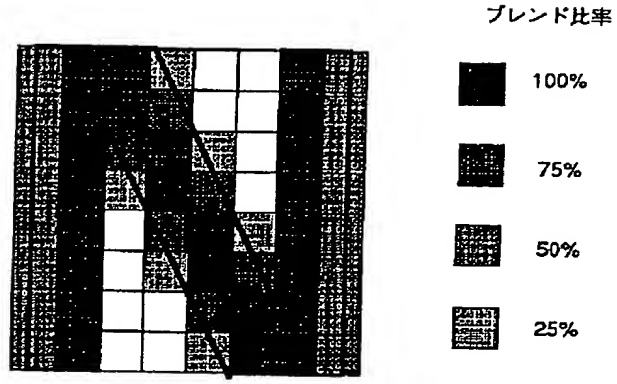
【図６】

図 6



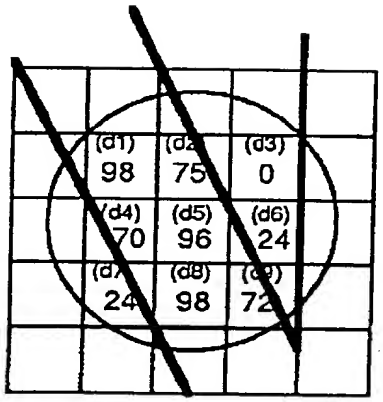
【図８】

図 8



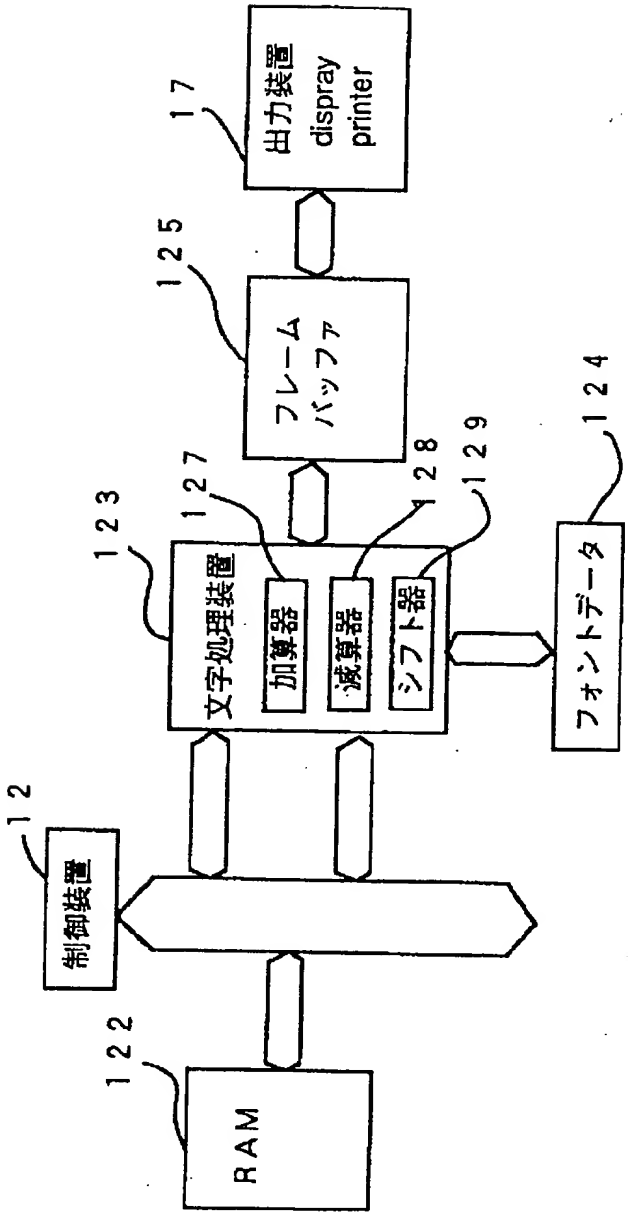
【図１２】

図 12



【図７】

図 7


出力装置
dispray
printer
17
125
フレーム
バッファ
127
128
129
124
123
文字処理装置
加算器
減算器
シフト器
フォントデータ
12
制御装置
122
RAM

【図１０】

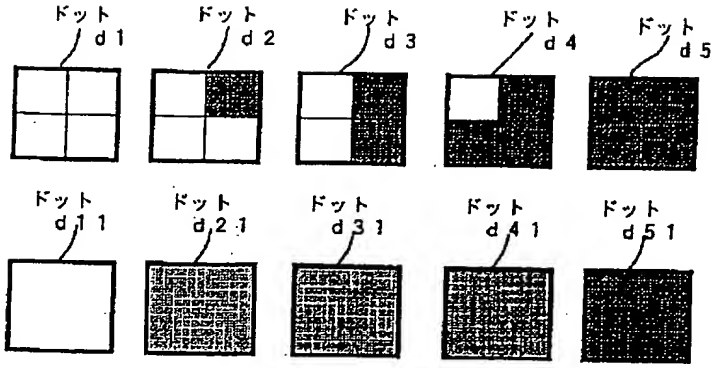
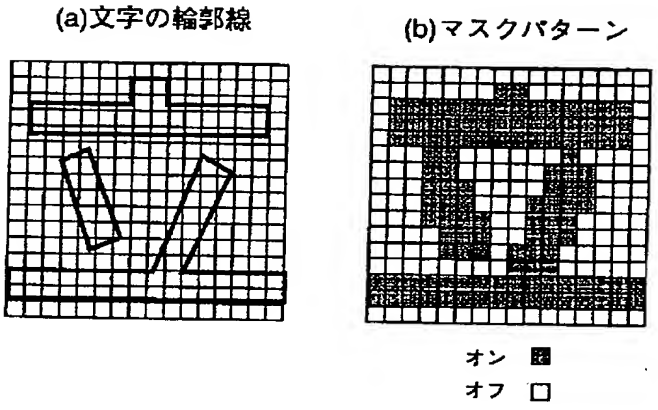


図 10

【図１３】

図 13



【図１４】

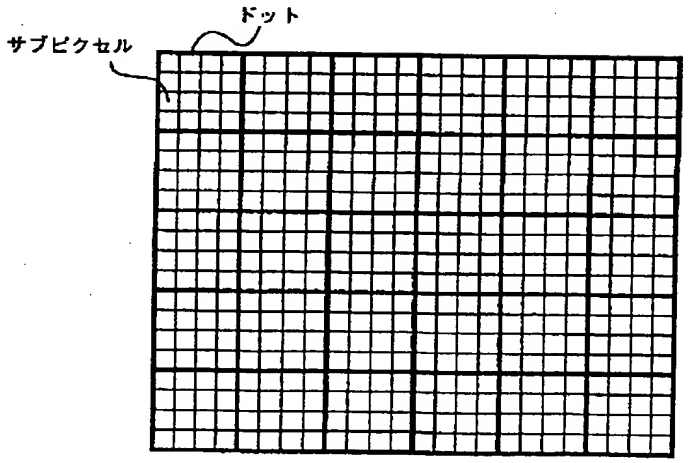
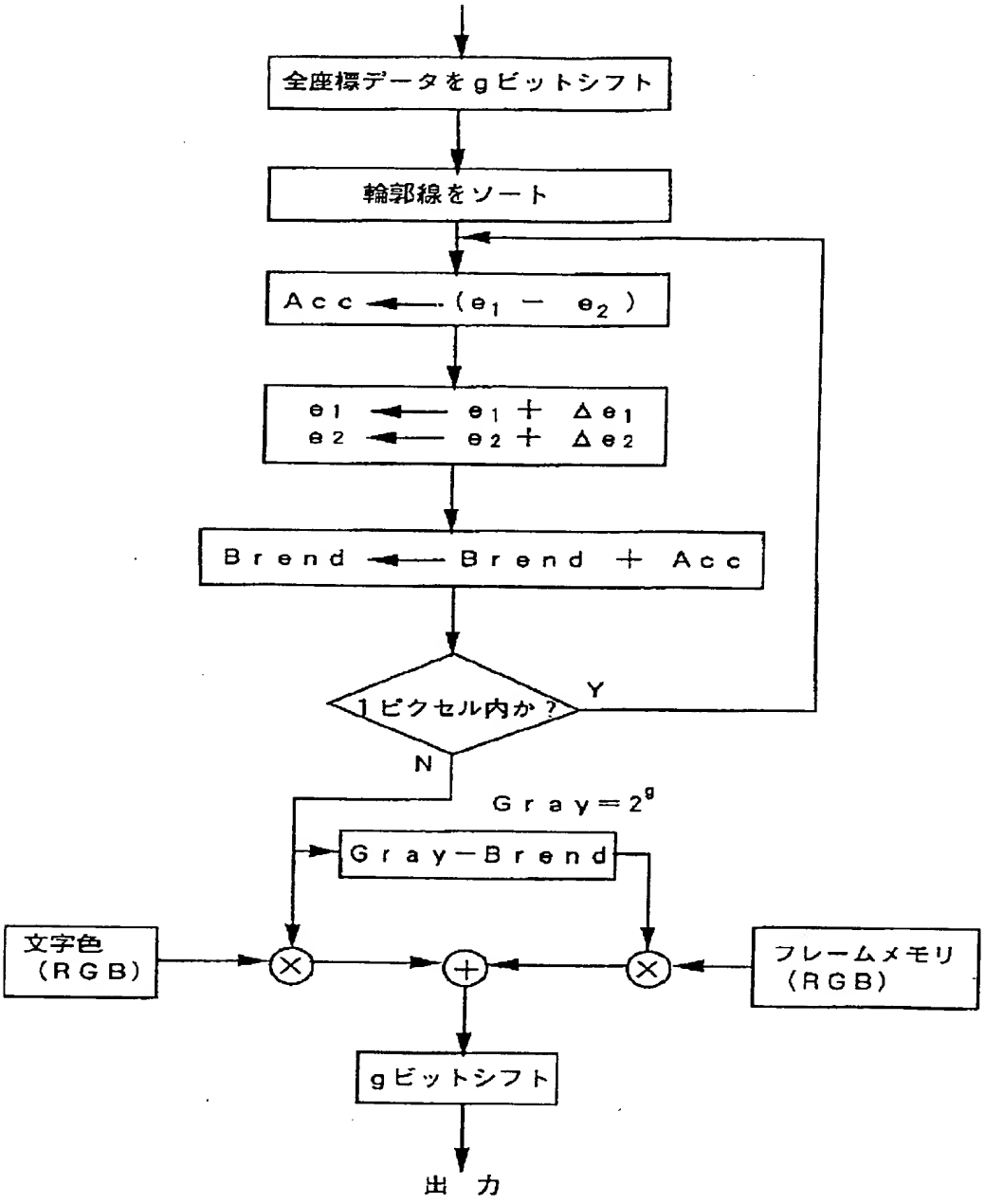


図 14

【図１１】

図 11


全座標データをｇビットシフト
輪郭線をソート
Ａｃｃ ← (e1 − e2)
e1 ← e1 ＋ Δe1
e2 ← e2 ＋ Δe2
Ｂｒｅｎｄ ← Ｂｒｅｎｄ ＋ Ａｃｃ
1ピクセル内か？
Y
N
Ｇｒａｙ＝2^g
Ｇｒａｙ−Ｂｒｅｎｄ
文字色
（ＲＧＢ）
フレームメモリ
（ＲＧＢ）
ｇビットシフト
出　力

【図１５】

図15